How to Open a New Library

Version 20.0 and later

CONFIDENTIAL INFORMATION

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

Document released: May 21, 2013

Web address: http://www.exlibrisgroup.com

Table of Contents

1 Overview

This document describes how to open a new ALEPH library with the `open_new_library` program.

Opening a new library in ALEPH means creating a new working environment for a certain ALEPH data type (for example, bibliographic data, administrative data, and so on).

Opening a new library is done by copying the entire environment (excluding data) of a similar existing ALEPH library (this source library is referred to as the "template library" in this document) and modifying the environment setup and configuration to the needs of the new library.

The environment includes: a Unix directory structure, an Oracle schema, and some definitions in alephe configuration tables. The program open_new_library will create the Unix directory structure and the Oracle scheme, and modify the required basic setup and configuration files in the library directory structure. The required modifications in the alephe configuration files must be done manually.

2 Preparations: Actions before Program Activation

2.1 Choosing Library Name

ALEPH libraries have a user-specific five-character code. Following an ALEPH naming convention, the code is made up of three characters followed by two digits, where the digits identify the library data type. For example, a Bibliographic (BIB) library contains bibliographic records and is identified by a number between 01-09 (for example, xyz01).

The following conventions are used for library types:

xxx00 – for USR00 library type.

xxx01-xxx09 - for bibliographic records.

xxx10-xxx19 – for authority records.

xxx40-xxx49 – for ILL library (bibliographic records of the requests).

xxx30-xxx39 – for Course Reading library (for course reserve materials).

xxx50-xxx59 – for administrative records library (Items, Circulation, Acquisitions, Serials).

xxx60-xxx69 – for holdings records.

Examples:

xyz02 is a BIB library.

xyz11 is an AUT library.

xyz52 is an ADM library.

The ALEPH naming convention is just a convention; it is *not* a system requirement. However, as stated, the open_new_library program requires that the naming convention be followed, and if a mismatch is found between the new library name and the ALEPH naming convention, a warning message will be given.

In addition, the first three characters of a user library must NOT be one of the reserved ALEPH library names, namely usm, uni, mab, ext, utf, vir, or usr.

2.2 Choosing the Template Library

The template library is either an ALEPH demo library or an existing user library.

The template library should be of the same data type as the new library, and if possible, its setup should be as close as possible to the new library.

The data type of the template library will determine the data type of the new library. If the data type of the template library cannot be determined because its name is not according to the convention, then the new library data type will be determined by the name of the new library. If both names are not according to the convention, then the program will assume that the new library type is BIB.

A warning message will be given in all case were the names are not according to the convention or the libraries' data type do not match.

The ALEPH demo libraries are named after the cataloging standards: USMnn, UNInn, and MABnn (for MARC 21, UNIMARC, and MAB, respectively).

The following are the ALEPH standard demo libraries:

USR00: For usr (users - patrons and staff), pw (password), and z105 libraries

USM01; UNI01; MAB01 for bibliographic records libraries

USM10; UNI10; MAB10/11/12 for authority records libraries

USM40; UNI40; MAB40 for ILL records libraries (bibliographic records of the ILL request)

USM30; UNI30; MAB30 for Course Reading libraries (course reserve materials)

USM50; UNI50; MAB50 for administrative records libraries (Items, Circulation, Acquisitions, Serials)

USM60; UNI60; MAB60 for holdings records libraries

Please note that if the library is an ADM library it can also be used as the usr library and/or the pw library and/or the Z105 library.

If the library being created is intended to serve only as pw library and/or usr library and/or Z105 library, the template library should be USR00.

2.3 Choosing the Disks (and Path) for Library Directories

An ALEPH library environment includes a Unix directories tree of setup files, configuration tables directories, and data files directories.

It is recommended that large libraries separate the setup and configuration directories and the data directories. By default, all the directories are under the same root ($data_root) under the “u-tree”.

If the directories are separated, it is recommended that setup and configuration directories be on the same device as ALEPH, and that the large data directories, $data_files and $data_scratch, be on another disk and not under $data_root. In such cases, $data_scratch and $data_files directories will be physically placed under the x_path (the name of the path will be supplied by the user), but a symbolic link will be program-generated so that they are logically placed under $data_root of the new library.

Note: Open_new_library script does not work with Upper case path.

2.4 Adding Definitions to aleph_start.private

`aleph_start.private` is a configuration file that contains definitions of ALEPH libraries and directories. It also contains logical assignments (the only place with physical references to ALEPH directories). It is invoked by every ALEPH process.

To add definitions to aleph_start.private

1. Open `aleph_start.private` for editing by entering the following in the command line:

```
vi $alephe_root/aleph_start.private
```

2. Add the library’s name to the following environment variables:

Note
xxxnn means <library name><library number>. The library name must be in lowercase. For example, `dem01`, `dem50`, and so on.

All libraries:

```
setenv  ALEPH_LIBS              "$ALEPH_LIBS xxxnn"
setenv  QUE_STARTUP_LIBS        "$QUE_STARTUP_LIBS xxxnn"
```

If the library’s data type is BIB (xxx01-09) or ILL (xxx40-49) or Course Reading (xxx30-39):

```
setenv  WORD_STARTUP_LIBS       "$WORD_STARTUP_LIBS xxxnn"
```

If the library’s data type is BIB (xxx01-09):

```
setenv  ACC_AUT_STARTUP_LIBS   "$ACC_AUT_STARTUP_LIBS xxxnn"
```

If the library’s data type is ADM (xxx50-59):

```
setenv  REQUEST_STARTUP_LIBS   "$REQUEST_STARTUP_LIBS xxxnn"
```

If needed, add the library's name to other <list>_STARTUP_LIBS lists, such as RLIN, SIP2, SLNP, and so on.

3. Define the path to library root directory:

```
setenv xxxnn_dev
```

<the library basic path; that is, the path where the library is located>

For example:

```
setenv      xxxnn_dev    /exlibris/aleph/u16_1
setenv      yyynn_dev    /exlibris2/aleph/u16_1
```

or by using other predefined environment definitions:

The definitions of USER_MOUNT, ALEPH_VERSION, and ALEPH_COPY are defined in $alephe_root/aleph_start.

For example:

```
setenv    USER_MOUNT              /exlibris/aleph
setenv    ALEPH_VERSION           16
setenv    ALEPH_COPY              1

setenv mgm01_dev ${USER_MOUNT}/u${ALEPH_VERSION}_$ALEPH_COPY
```

2.5 Changing Definitions in aleph_start

If the new library will serve as the pw library and/or usr library and/or z105 library, modification of the definitions in aleph_start are required:

Open `aleph_start` for editing by entering the following in the command line:

```
vi $alephe_root/aleph_start
```

and add/change the definitions for the pw, usr, and z105 libraries.

For example:

```
setenv    pw_library      XYZ50
setenv    usr_library     XYZ50
setenv    z105_library    XYZ50
```

Please note that the library name in these definitions must be in uppercase.

2.6 Choosing Library Size and Creating Default Tablespaces

The space allocated for the Oracle tables and the tablespaces to be used are defined in the file list. You must choose the appropriate file_list_size from the following choices according to the expected number of Z00 records (that is, doc records) that will be stored in the library:

DEMO – To be used only by the demo libraries

VSMALL - Up to 10,000 documents

SMALL - Up to 100,000 documents

MEDIUM - Up to 1,000,000 documents

LARGE - Up to 5,000,000 documents

HUGE - Larger than 5,000,000 documents

The program will configure the system to use an appropriate default file_list.

Please ensure that the tablespaces defined in the default file_list exist before the new library is created (unless the 6th parameter in the program activation is N; see *Create Oracle User (Optional)* on page 11).

To check what the default tablespaces are, and if they exist, use UTIL/O/17/3. To create the required tablespaces, use UTIL/O/17/1.

3 Activation: Creating a New Library Based on a Template Library

3.1 Syntax

The syntax of the 'open_new_library' command is as follows:

```
csh -f $aleph_proc/open_new_library <Library_Name> <File_List_Size>
<Template_Library> <X_Path> <USR_Library> <Create_Oracle_User>
```

3.2 Activation Modes

There are two activation modes for opening the library:

Batch Mode

Batch mode – provide all 6 parameter values. To use default parameters, use a null value (that is, "") as a parameter value. If you enter all six parameters, the system will not prompt you to enter values.

Interactive Mode

If you enter fewer than six parameters, the program will work in interactive mode and you will be prompted for each of the missing parameters. The last three parameters have default values, given in square brackets. If you want to accept the default value, press <enter>.

Note

If the program exits in the middle of the process due to error/s, the new library, together with any content that was already created, will be deleted.

3.3 Key to Parameters

3.3.1 Library_Name (Mandatory)

The name of the library to be opened.

The program checks the following:

Does the library already exist?

Is it a reserved library name? (is it usm, uni, mab, ext, utf, vir, or usr?)

Is it defined in ALEPH_LIBS? (in $alephe_root/aleph_start.private)

Is <library name>_dev defined? (in $alephe_root/aleph_start.private)

Does the Unix user have permission to create the root directory of the library?

3.3.2 File_List_Size (Mandatory)

Size of library to be set in $data_root/prof_library.

The program checks the following:

Is the value supplied a legal value for `file_list_size`: DEMO, VSMALL, SMALL, MEDIUM, LARGE, HUGE?

Do the tablespaces defined in the appropriate file_list exist? (Note: this check is disabled if the 6th parameter is N.)

3.3.3 Template_Library (Mandatory)

The library to be used as a template.

The program checks if the library exists and if so, it checks that it is not the name of the library being created.

The program will:

Copy all tables/files/directories and definitions.

Replace all occurrences of the template library with the new library name, in all the copied files.

Create a local `path_convert` file if it does not exist in the template library.

Create proper definition of file_list_type in $data_root/prof_library file.

Add the following commented-out lines in $data_root/prof_library:

```
#        setenv   file_list_TS1D        ts_XXXXX_1D
#        setenv   file_list_TS2D        ts_XXXXX_2D
#        setenv   file_list_TS3D        ts_XXXXX_3D
#        setenv   file_list_TS4D        ts_XXXXX_4D
#        setenv   file_list_TS1X        ts_XXXXX_1X
#        setenv   file_list_TS2X        ts_XXXXX_2X
#        setenv   file_list_TS3X        ts_XXXXX_3X
#        setenv   file_list_TS4X        ts_XXXXX_4X
```

Set all loop_length definitions of the various batch services in $data_root/prof_library to 50,000.

3.3.4 X_Path (Optional)

Path for $data_files and $data_root directories when these are separated from the setup and configuration directories (see *Choosing the Disks (and Path) for Library Directories* on page 6). Default is Space or Null.

By default, the physical directories `$data_files` and `$data_scratch` will be under `$data_root`.

If the value is not null, the program will create these directories in the given location and replace the appropriate physical directories under `$data_root` by an appropriate symbolic link.

3.3.5 USR_Library (Optional)

Indicates whether the library will be function also as one or more of the pw/usr/z105 libraries. The default is N (No) unless the template library is USR00, in which case the default is Y (Yes).

Since any library may also function as one or more of pw/usr/z105 libraries, the user can supply a list - one or more of usr, pw, z105, separated by comma, with no spaces. Y is equivalent to usr,pw,z105.

If the library being created should function as one or all of the pw/usr/z105 libraries, the program checks whether the usr/pw/z105_library/ies defined in $alephe_root/aleph_start are the same as the library name (in uppercase) currently being opened. The check ensures that the file_list will include the definitions for the additional pw/usr/z105 functionalities.

If the template library is usr00, then the USR_library parameter cannot be “N”. If the user enter "N" as the USR_library parameter, then the program will ignore the “N” and will act as “Y” was given.

3.3.6 Create Oracle User (Optional)

Whether or not to drop and create Oracle user for the library. Default is Y (Yes).

If set to Y, the Oracle user (if it already exists) will be dropped and a new Oracle environment will be created according to the file_list – Oracle user, tables, indexes, synonyms, triggers, and sequences.

If the Oracle user was not created (that is, the program was activated with N), then the Oracle environment must be manually created later (using UTIL/O/9/2), before starting to work in the library.

3.4 Examples

```
csh -f $aleph_proc/open_new_ library tst01  SMALL usm01  “”  N  Y
```

Opens the BIB library tst01 using the demo library usm01 as the template library.

The library is configured to store up to 100,000 BIB records.

$data_file and $data_scratch directories will be located under $data_root

The library will not be used as one of the usr, pw, or z105 libraries.

Oracle user and objects will be created.

```
csh   -f   $aleph_proc/open_new_library     tst50     HUGE     xyz50
/exlibris2/aleph/u16_1  pw,usr  Y
```

Opens the ADM library tst50 using the user library xyz50 as the template library.

The library is configured to store more than 5,000,000 ADM records.

$data_file and $data_scratch directories will be located under /exlibris2/aleph/u16_1

The library will be used as the usr and pw libraries, but not as the z105 library.

Oracle user and objects will be created.

4 Actions after Activation

You must log out and then log in before you proceed with the following actions in order to implement all the changes and definitions that were made in the environment, and to perform:

dlib xxxnn

4.1 Synchronize Library File Headers

It is recommended to use UTIL/H/2 to ensure that all headers of the new library tables are up to date.

4.2 Oracle Tables and Tablespace (file_list and prof_library)

The script creates an empty $data_root/file_list (that is, the file is created with no content). In this case all Oracle objects will have the default attributes. If the library requires a different value for a specific Oracle object, it must be added manually to the `file_list` to override the default value.

Open `file_list` for editing by entering the following in the command line:

```
dlib xxxnn
dr
vi $alephe_file_list
```

If you want to use tablespaces other than the default, you must edit prof_library.

Open `prof_library` for editing by entering the following in the command line:

```
dlib xxxnn
dr
vi prof_library
```

and modify the following commented lines accordingly:

```
#       setenv   file_list_TS1D       ts_XXXXX_1D
#       setenv   file_list_TS2D       ts_XXXXX_2D
#       setenv   file_list_TS3D       ts_XXXXX_3D
#        setenv   file_list_TS4D        ts_XXXXX_4D
#        setenv   file_list_TS1X        ts_XXXXX_1X
#        setenv   file_list_TS2X        ts_XXXXX_2X
#        setenv   file_list_TS3X        ts_XXXXX_3X
#        setenv   file_list_TS4X        ts_XXXXX_4X
```

XXXXX can be any valid name. Please note that the same tablespace can be shared by several libraries.

Do not forget to remove the # in the beginning of the line.

In addition, you must ensure that these tablespaces exist (using UTIL/A/17/12) and, if necessary, create them (using UTIL/O/17/1).

If you have modified the `file_list`, or the tablespaces to be used by the library, you must activate UTIL/A/17/1/all to make the changes effective.

4.3 loop_length

The program sets all the loop_length variables (defined in prof_library) to 50,000.

If you want to use different values, you must edit prof_library.

The full list is taken from the demo library usm01. If this library does not exist, then the full list is taken from the template library. In this case, check to confirm that the template library includes the full list.

Open `prof_library` for editing by entering the following in the command line:

```
dlib xxxnn
dr
vi prof_library
```

4.4 Updating Configuration Tables

The following tables should be modified as appropriate:

4.4.1 library_relation

This table defines links between libraries. The system requires that the relationship between the BIB/ADM/HOL and ILL libraries be defined.

Open `library_relation` for editing by entering the following in the command line:

```
vi $alephe_tab/library_relation
```

Example of the `library_relation` table:

```
!1   2     3     4     5     6
!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!>
ADM USM01 USM50 USM51
ADM USM40 USM50
ADM USM30 USM50
BIB USM50 USM01 USM40 USM30
BIB USM51 USM01
```

```
        HOL USM50 USM60
        ILL USM50 USM40
```

4.4.2 tab_sub_library.lng (relevant only for ADM libraries)

The `tab_sub_library.lng` table defines the sublibraries of all the ADM libraries on the server.

Open `tab_sub_library.eng` for editing by entering the following in the command line:

```
        vi $alephe_tab/tab_sub_library.eng
```

If your system is configured for languages other than English, the additional copies of the tab_sub_library (for example, tab_sub_library.fre) must also be updated.

Example of the `tab_sub_library.eng` table:

```
USM50 2 USM50 L Exlibris Demo Libraries                 USM50       USM50 ALEPH
MED   1 USM50 L Medicine Library                 15H    CHE   17A   MED   ALEPH
HYL   1 USM50 L East Asian Library               HYL    HYL   17A   HYL   ALEPH
HIL   1 USM50 L Humanities Library               HIL    HIL   17A   HIL   ALEPH
LAM   1 USM50 L Main Undergraduate               LAM    LAM   17A   LAM   ALEPH
WID   1 USM50 L Main Library                     WID    WID   WID   WID   ALEPH
HILR  4 USM50 L Humanities Reading Room          HIL    HIL   17A   HIL   ALEPH
```

4.4.3 tab_base.lng

The `tab_base.lng` table defines the logical and physical databases that can be accessed by the user through the Web OPAC and search functions.

Open `tab_base.eng` for editing by entering the following in the command line:

```
        vi $alephe_tab/tab_base.eng
```

If your system is configured for languages other than English, the additional copies of the tab_sub_library (for example, tab_base.fre) must also be updated.

Example of the `tab_base.lng` table:

```
USM01                   Ex Libris University                 USM01 N
WID                     Main Library          USM01          USM01 Y wsl=wid
HIL                     Humanities Library    USM01          USM01 Y wsl=hil
HYL                     East Asian Library    USM01          USM01 Y wsl=hyl
LAM                     Main Undergraduate    USM01          USM01 Y wsl=lam
MED                     Medicine Library      USM01          USM01 Y wsl=med
GEEDU                   Main/Undergrad.       USM01          USM01 Y wsl=(wid or lam)
LALIT                   Law/Economics         USM01          USM01 Y wsl=(law or lit)
SERIALS                 Serial (USM01)                       USM01 N wfm=SE
MONOGRAPHS              Monographs (USM01)                   USM01 Y wfm=BK
RUSSIAN                 Russian    (USM01)                   USM01 N wln=RUS
```

4.4.4 base-list (relevant only for BIB libraries)

This HTML file contains the list of databases available to the Web OPAC user. It is located in `$alephe_root/www_f_eng/base-list`.

Open `base-list` for editing by entering the following in the command line:

```
wf (this puts you into the www_f_eng directory)
vi base-list
```

If your system is configured for languages other than English, the additional copies of the base-list (for example, www_f_fre/base-list) must also be updated.

Example of a library in the `base-list` file:

```
<td class=text3>
   <img src="&icon_path/f-small-dot.gif" border="0" alt="">
   <a href="&server_f?func=find-b-0&local_base=LOC">
   Library Of Congress - (Z39.50) </a>
</td>

<td class=text3>
   <img src="&icon_path/f-small-dot.gif" border="0" alt="">
   <a href="&server_f?func=find-b-0&local_base=monographs">
   Monographs Catalog (USM01)</a>
</td>
```

4.4.5 bib/pc_tab/catalog/scancode.dat (relevant only for BIB libraries)

This table defines the selection of libraries and the headings file of each one that is accessed when the Search Headings" function is invoked in the Cataloging module.

Steps

1. Open scancode.dat for editing by entering the following in the command line:

   ```
   dlib <lib_name>

   dr

   cd pc_tab/catalog/

   vi scancode.dat
   ```

2. In the line LOCAL replace, if necessary, the authority libraries used by the template library by the appropriate authority libraries.

   For example:

   ```
   LOCAL  XYZ01  USM10  USM12  XYZ01  USM11
   ```

   Will be modified to

   ```
   LOCAL  XYZ01  XYZ10  XYZ12  XYZ01  XYZ11
   ```

3. Activate UTIL/M/7 to Update Tables Package

4.4.6 bib/tab/tab_locate

This table defines procedures for locating a record in other libraries

Steps

1. Open tab_locate for editing by entering the following in the command line:

```
dlib <lib_name>
dt
vi tab_locate
```

2. Type in your authority library codes instead of the template's authority libraries' codes.

4.4.7 bib/tab/tab_aut (relevant only for BIB libraries)

This table defines among others which authority database should be checked for a match.

Steps

1. Open tab_aut for editing by entering the following in the command line:

```
dlib <lib_name>
dt
vi tab_aut
```

2. Type in your authority library codes instead of the template's authority libraries' codes.

4.4.8 XXXnn/tab/tab_z105

This table defines the settings for sending a message (Z105) from one library to another every time a record is added/updated.

Steps

1. Open tab_z105 for editing by entering the following in the command line:

```
dlib <lib_name>
dt
vi tab_z105
```

2. Replace the template's USM library codes (USMnn) with your library codes.
   In the examples below, replace ABC01, ABC10, ABC30, and ABC50 with
   your library codes:

- If an Authority library called ABC10 is added, in **abc10**/tab/tab_z105,
  change the following from:

```
UPDATE-DOC 4 USM01
```

to:

```
UPDATE-DOC 4 ABC01
```

- If a Bibliographic library called ABC01 is added, in **abc01**/tab/tab_z105, change the following from:

  ```
  UPDATE-DOC        e USM30
  INDEX-DOC         e USM30
  ```

  To:

  ```
  UPDATE-DOC        e ABC30
  INDEX-DOC         e ABC30
  ```

- If an Administrative library called ABC50 is added, in **abc50**/tab/tab_z105, change the following from:

  ```
  UPDATE-SDI        8 USM01
  UPDATE-Z30        1 USM01
  UPDATE-ITM        m USM50
  ```

  To:

  ```
  UPDATE-SDI        8 ABC01
  UPDATE-Z30        1 ABC01
  UPDATE-ITM        m ABC50
  ```

Note:
Instead of changing the tab_z105 lines from USM to ABC, you can delete the lines that describe functions that are not used at your site.
For example:

- If you do not have a Course Reading library associated with a BIB library, the XXX30 lines can be removed from the BIB library table.
- In the ADM library table, UPDATE-Z30 line can be removed if you are not a German library.

See the tab_z105 lines table header for more information about the specific functionality of each line.

5 Checking Your New Library

To ensure that the library was opened correctly

1. Log out, and then log in. This implements all the changes / definitions that were made in the configuration tables.
2. From the command line, enter:

   ```
   dlib <lib_name>
   dr
   ls
   ```

3. Check if the library contains the directories copied from the template library.
4. Check if all occurrences of the template library in `prof_library` were replaced with the current one.
5. If you have selected to create an Oracle user, enter the following in the command line:

```
s+ <lib_name>
```

   You should get the prompt:

```
SQL-<lib-name>
```

   Then enter:

```
SQL-XXXNN>desc z00;
```

   You should get:

```
Name                                      Null?    Type
----------------------------------------- -------- --------------------
--------
Z00_DOC_NUMBER                            NOT NULL CHAR(9)
Z00_NO_LINES                                       NUMBER(4)
Z00_DATA_LEN                                       NUMBER(6)
Z00_DATA
```

6. Exit by entering:

SQL-<lib-name> > exit